PI-CONNECT OPC+ Interface

incl. Option Archive Recovery

User manual

V3.8a

Siemens PI-CONNECT OPC+ Interface
to Plant Information System PI®
of OSIsoft, LLC

Siemens AG,GER I CS MTE MHM

Version: V3.8a dated April, 30th 20134

Contact:

Telephone	(0621) 456-2303
Fax	(0621) 456-3334
Email	info.rhm.industry@siemens.com
Address	Siemens AG GER I CS MTE MHM, Hr. Lutter Dynamostr. 4 D - 68165 Mannheim Germany

Subject to change without further notice / Änderungen vorbehalten

© 2003-2014 Siemens AG, GER I IS MTE OC MHM. All rights reserved.
Dynamostr. 4, D-68165 Mannheim, Germany

Trademarks:
PI is Trademark of OSIsoft, LLC
Microsoft EXCEL, Microsoft Windows NT and Microsoft Windows XP, Windows 7 are Trademarks of Microsoft Corporation.

Contents

1. Overview 5
   1.1 General 5
   1.2 Exemplary configuration 6
   1.2.1 PCS7 via Open PCS7 7
   1.2.2 PCS7 / WinCC via WinCC OPC server 8
   1.3 Performance 9
   1.4 Features of the interface 10
   1.5 Option Archive Recovery (via OPC HDA) 10
   1.6 Licensing 10

2. Install und Uninstall 11
   2.1 Requirements for the Installation 11
   2.2 Installation of the Interface Software 11
   2.3 Uninstall 11

3. Configuration 12
   3.1 Structure of the file SiOPC_IF.ini 12
   3.2 Licence settings 13
   3.2.1 Creating a licence 13
   3.2.2 License Activation 13
   3.2.3 Emergency function 13
   3.3 General settings 14
   3.4 PI System Parameters 15
   3.5 OPC DA / OPC HDA Parameters 15
   3.6 Global Parameters (for all instances valid) 17
   3.6.1 Replacement tables for digital conditions (DigiCodeTable) 17

4. Starting and terminating of the Interface 18
   4.1 Starting as Windows console application 18
   4.2 Simplified start as Windows console application by linkages 18
   4.3 Automatic start up as console application after Login 19
   4.4 Shutdown of the console application 19
   4.5 Starting as Windows Service 20
   4.5.1 General 20
   4.5.2 Register, configure and operation of an Instance as a service 20
   4.5.3 Removing service of an instance 21

5. Function of the Siemens OPC+ Interface 22
   5.1 Scope 22
   5.2 Connecting to redundant PCS7 OS (redundant OPC server) 22
   5.3 Configuration of acquisition cycles 23
   5.4 Special feature on time stamping 24
   5.5 Writing PI data “OnDataChange” 25
   5.6 Bit masking 26
   5.7 Scaling 26
   5.8 Replacement tables for digital conditions 26
   5.9 Archive Recovery OPC HDA (optional) 27
   5.9.1 General and requisites 27

5.9.2 Functionality 27
5.9.3 Performance 28
5.9.4 Archive recovery system load 28
5.10 Supported Data Types and their conversion 29
5.10.1 List of supported Data Types 29
5.10.2 Data Type Conversion for reading Points 29
5.10.3 Data Type Conversion of writing points 30
5.10.4 Writing to OPC Variables of Data Type VT_BOOL 30
5.11 PI Point States used by the interface 31
5.12 HDA Window Recovery Mode 32
5.13 Logfile limitation 32
5.14 Autostop Function 32
5.15 Operation on missing of OPC Server call-backs 33
5.16 Controlling of the OPC connection via an OPC test Item 33
5.17 Option Connection specific Item Prefix / Suffix 33
5.18 Performance Tags 34
5.19 OpenPCS 7 Workaround 35

6. PI Point configuration 36
6.1 Automatic configuration tool (option) 36
6.2 Siemens OPC+ Interface specific attributes 36
6.3 PointSource Attribute 36
6.4 ExDesc attribute 37
6.4.1 Structure 37
6.4.2 OPC ItemID (PCS7 variable name) 37
6.4.3 OPC Access Path 37
6.4.4 Bit mask and Shift parameter 37
6.5 Location1 Attribute (Instance) 38
6.6 Location2 Attribute (Scaling on/off) 38
6.7 Location3 Attribute (IO-Type) 38
6.8 Location4 Attribute (Scan Class) 38
6.9 Location5 Attribute (PCS7 / WinCC Archive) 38
6.10 InstrumentTag Attribute (name of the archive variable) 39
6.11 UserInt1 (Number of replacement table) 39
6.12 Userreal1 (scaling factor) 39
6.13 Userreal2 (scaling offset) 39
6.14 SourceTag Attribute 39

7. Hints on PCS7, WinCC and the WinCC OPC Servers (DA and HDA) 40
7.1 General 40
7.2 Access to the OPC Server, Remote access via DCOM 40
7.3 Identification of the WinCC Process value archives 41
7.4 PCS 7 Delta Loading 42

8. Hints for API Installation and PI Server Configuration 43
8.1 PI-interface SDK API Software installation 43
8.2 PI User for the PI-CONNECT OPC+ interface 43
8.3 PI-Trust entry on the PI server 44
8.4 Running the Interface against a high available PI Server (HA-PI) 45

9. Revision History 47

1. Overview

1.1 General

The Siemens PI-CONNECT OPC+ Interface reads process data from a Process Control System (stand alone or redundant pair) via OPC Data Access (OPC DA) and sends this data to the PI system. Writing data out of the PI system to variables of the Process Control System is also supported.

The PI-CONNECT OPC+ Interface provides the option „Archive Recovery“. See also section 1.5 Option Archive Recovery (via OPC HDA). For best performance the Interface uses OPC HDA only to close data gaps. In online mode the Interface uses OPC DA. On startup of the PI-CONNECT OPC+ interface or after a reconnect of the connection to the process control system the data gap can be closed out of the archives of the process control system.

For the option „Archive Recovery“ the access to the archives of the process control system via the OPC Historical Data Access (OPC HDA) server is necessary.

In the online mode (using OPC DA mode), the process data transfer between the DCS and the PI interface will run only on data changes. Due to that the DCS has a low load. For the cyclical supply of the PI exception test (check for horizontal value changes), events with current time stamps were built (*see section 5.4 Special feature on time stamping*).

In the PI-CONNECT OPC+ interface a mechanism is implemented, which can handle the switch over of the two redundant OPC servers itself (option).

The special features of PCS7 / Win CC OPC DA server (Time stamp correction, launch behaviour and status handling) were considered and prepared in PI for an optimized representation.

Further features of the PI-CONNECT OPC+ Interface are:

- The interface supports PI-HA
- Archive Recovery can be launched manually also in case of need (manual input of the time range which has to be filled out of the archives). See also chapter 5.12 HDA Window Recovery Mod.
- The optional PI-CONNECT Config Tool supports the users configuring and maintaining the tag configuration with PCS7 connection.
- To use large character strings in the PI tag configuration (e.g. for the OPC item name), PI-CONNECT OPC+ can use the PI SDK (USESDK configuration parameter).

Please regard for the installation of the PI-CONNECT OPC+ interface, that one instance of the interface connects one PCS7 server or server pair. That means for each server or server pair one instance of the interface has to be installed and licensed. For further information of the installation please refer to *chapter “2 Install und Uninstall”*.

In this description the functionality of the interface connecting OPC servers like in PCS7 or WinCC is explained. Using other DCS systems the user has to interpret parts of this description for his configuration.

The interface was designed for OPC DA V2.05a and OPC HDA V1.2 OPC servers. It is tested with OPC servers of the following systems:

- Siemens PCS7 V7 and V8 via OpenPCS7
- Siemens PCS7 V6.x, PCS7 V7.x, PCS7 V8 (via WinCC OPC DA server / WinCC Connectivity Pack)
- Siemens WinCC >V5.1, WinCC V6,x, WinCC V7.x
- Siemens Teleperm M OPC server (only DA)
- Siemens Simatic NET OPC (only DA)
- Siemens SPPA T2000 (Teleperm XP), SPPA T3000
- different OPC servers of other producers (on enquiry)

For hints of the installation and configuration of the PI SDK / PI API please read chapter *8: „Hints for API Installation and PI Server Configuration“* and the referring documentation of OSIsoft, LLC.

1.2 Exemplary configuration

PI-CONNECT OPC+ should be installed on an own PC. The interface is able to connect to both servers of a redundant server pair or to a stand alone server. Therefore the interface PC is connected by an own LAN card to the terminal bus of PCS7. By another LAN card the connection to the PI server is established. For communication with the PI server the interface software requires the PI API/SDK.

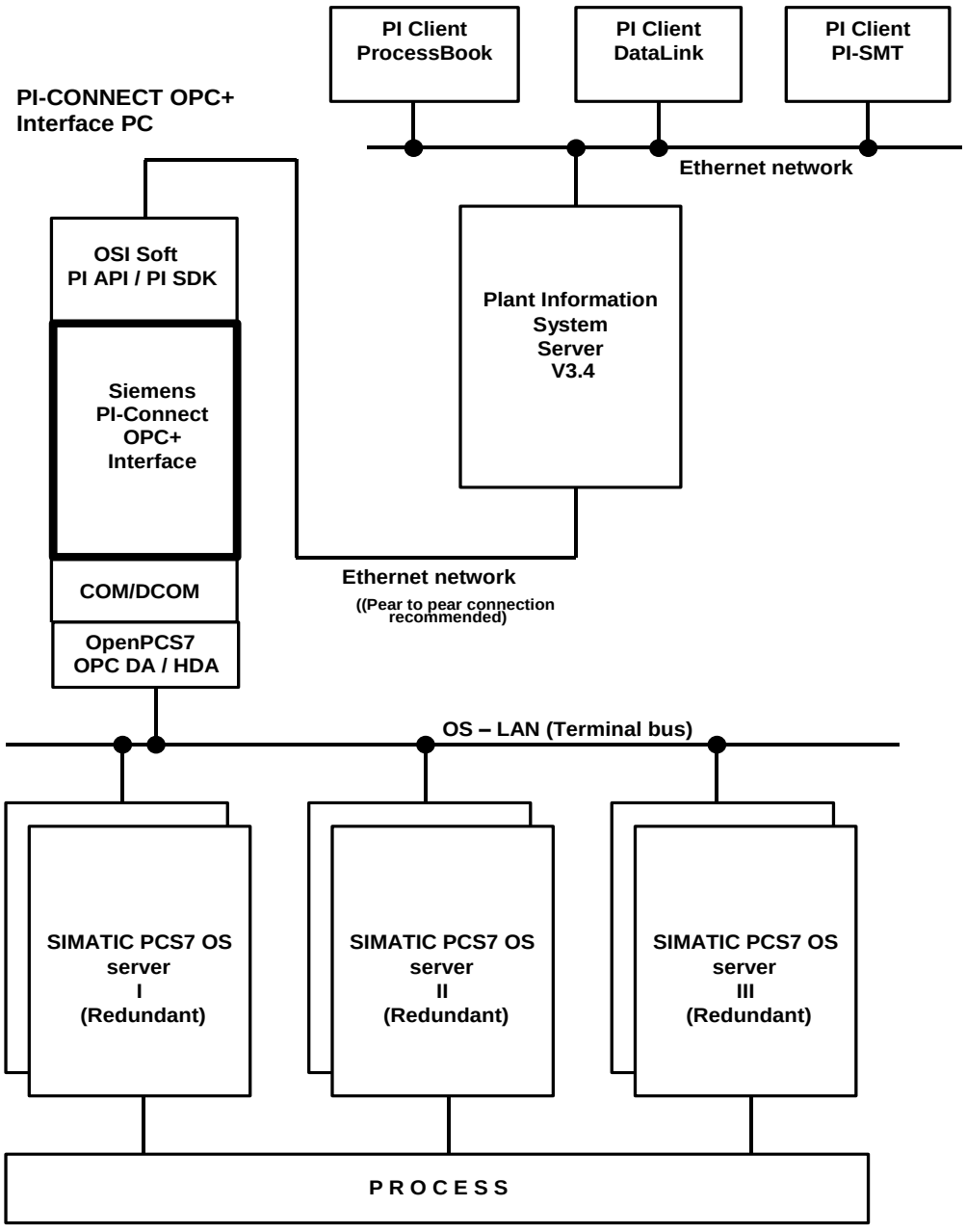
1.2.1 PCS7 via Open PCS7

For the process control system PCS7 V7 and V8 we recommend the additional interface component OpenPCS7 server installed on the interface PC. OpenPCS7 provides the OPC DA/HDA servers and hides the redundancy of the PCS7 OS servers. OPC HDA is used for the option Archive Recovery of the PI-CONNECT OPC+ interface.

In this configuration we also recommend the usage of one instance of the interface software for each PCS7 OS server pair (among others for the fault recognition).

OpenPCS7 is not included in the scope of supply of the PI-CONNECT OPC+ interface.

See also chapter 5.19 OpenPCS 7 Workaround.



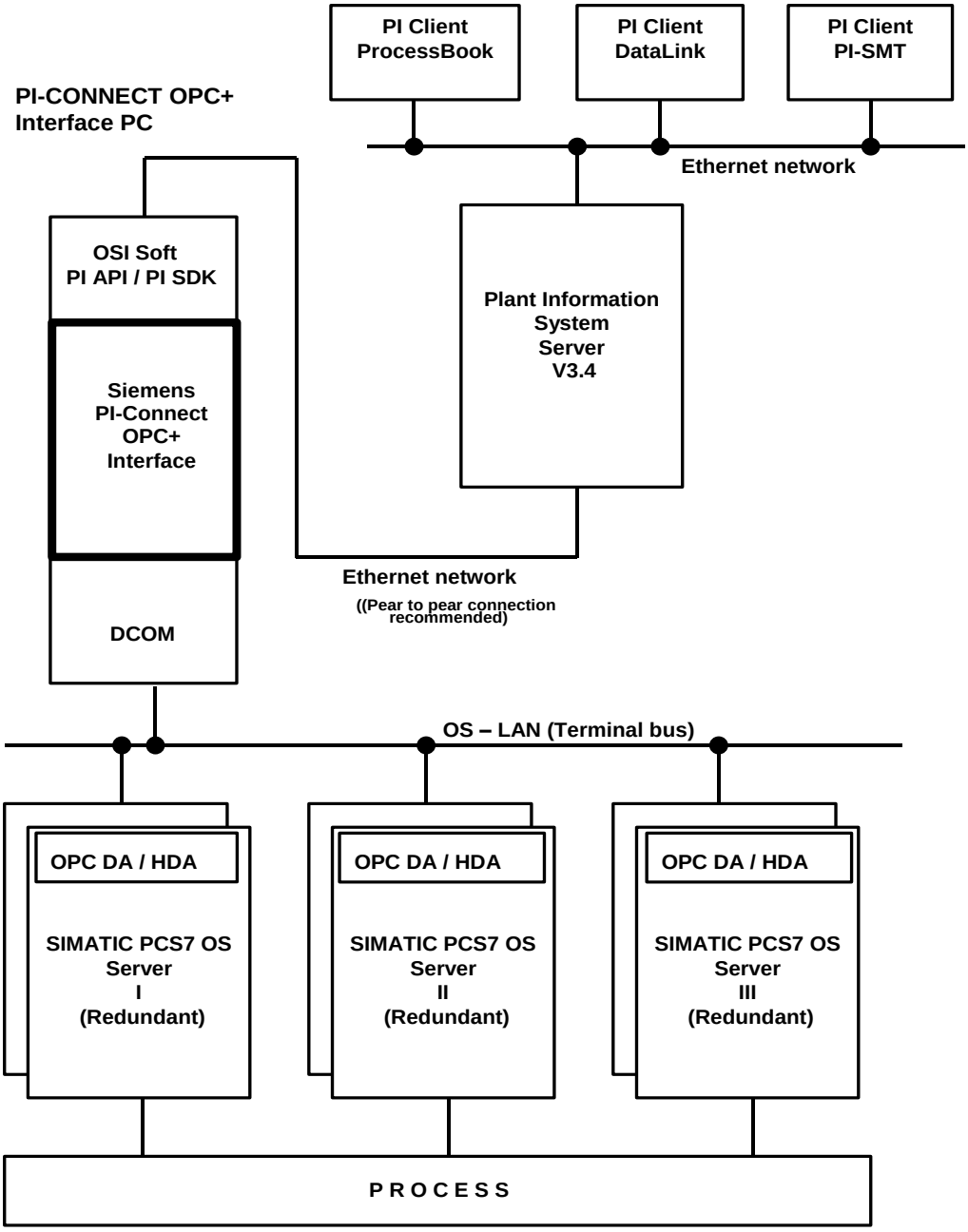
1.2.2 PCS7 / WinCC via WinCC OPC server

For WinCC and PCS7 V6 the PI-CONNECT OPC+ interface is able to handle the redundancy of the PCS7 OS servers or WinCC servers. The PI-CONNECT OPC+ interface uses the OPC DA server, which is part of every PCS7/WinCC OS server.

For the option Archive Recovery of the PI-CONNECT OPC+ on each PCS7 OS or WinCC server a license for the OPC HDA server is needed.

Therefore the WinCC Connectivity Pack assigned to its WinCC, or since PCS7 V7.1 the Open PCS7 for PCS7 OS clients on each PCS7 OS server can be used.

This configuration can also be used for PCS7 V7/8.



1.3 Performance

The PI-CONNECT OPC+ interface supports queued sending to PI (queued send to snapshot), API bode buffering und Auto-reconnect after connection loss to the PI server or to the process control system.

Redundancy of the PCS7 Systems is supported. Redundancy switching is done by the PI-CONNECT OPC+ interface (option).

Supported features:

Supported features	
Signup for Updates	Yes
Exception Reporting	Yes
Required software and licences from OSIsoft, LLC	Yes, PI Interface API/SDK. Please check conditions and terms with OSIsoft, LLC. As an option the PI Interface API/SDK can also be delivered by Siemens.
PI-SDK versions tested and recommended (PI-API is included)	PI SDK: V1.3.8 and V1.4.0
Required software and licences from the vendor of the DCS	Maybe OPC DA and OPC HDA server (has to be checked with the vendor of the DCS).
Operating systems tested by us	Windows XP, Windows 7 (x86 and x64), Windows 2003, Windows 2008, Windows 2008 R2
Multiple instances	Yes
Maximum number of points	Unlimited
Minimum cycle time	1 second
Parameter read (Inputs)	Cyclic, OnDataChange
Parameter write (Outputs)	OnDataChange
Time stamping	DCS or Interface PC
Archive Recovery	Yes as option, for PCS7 V6, also manual mode for disaster recovery available.
Addon Failover Online	Yes as option, but not in combination with Archive Recovery
Pointclass	Classic

1.4 Features of the interface

Using the recommended configuration *(refer to chapter1.2.1)* and a redundant PCS7 server a data rate of 2000 values per second could be achieved in a test. An Intel P4 PC 2,4 GHz has been used for the test.

Attention: The communication between the Automation system (AG) and the PCS7 server may be a bottleneck. In configurations with redundant AGs lower performance is possible. Also an overload of the channel has been detected. The spreading of the data rate to the separate AGs has to be kept in mind. Furthermore the existing load caused by the PCs7 project without the interface has to be considered. **Overload of the channel is not automatically shown by the PCS7 server!** Having data rates over 200 values a second and whenever the distribution on the AG´s is unknown it is recommended to activate the data collection step by step and to check the load of the channels with the possibilities of PCS7 servers. In addition we point to the PCS7 FAQs of the OPC performance issues.

1.5 Option Archive Recovery (via OPC HDA)

The option „Archive Recovery“ of the OPC+ IF is able to use OPC HDA to close the data gap since the last shutdown on the next start up (e.g. after service works on the interface PC or the network) or after the reconnect to the DCS. For further information please refer to *chapter 5.9,* Archive Recovery OPC HDA (optional).

Due to the needed OPC HDA server in the DCS, this option is available with version PCS7 V6 or higher. Archive recovery is an optional package. The license must comprise the option to enable the use of this functionality. Maybe you need in addition for the usage of the OPC HDA server additional licences for the Process Control System.

1.6 Licensing

- A license is necessary for each PI Interface PC. Each of these licenses allows connection to a single PCS7 Server pair.
- If a PCS7 process control system consists of more then a single PCS7 OS server pair a license extension is needed for each additional PCS7 OS server pair.
- If the PI-CONNECT OPC+ Interface shall handle itself switching between redundant OPC servers an according license is needed.
- Archive Recovery is an option for the PI-CONNECT OPC+ Interface which is needed only once for each Interface PC.
- Not redundant PCS7 OS servers as well as PCS7 OS Stations respectively redundant pairs of PCS7 OS Stations are treated like a redundant PCS7 OS server in by the PI-CONNECT OPC+ Interface.
- For each instance of the PI-CONNECT OPC+ Interface a license file is needed for correct operation. For generation of this license a tool is provided which must be run on the PI interface PC to collect the data necessary for the generation of the license file. Those data must be send to us by email. The license file is returned by email thereafter.
- Installation, test and evaluation of the PI-CONNECT OPC+ Interface may be done with the emergency licence provided with the delivery.(refer to section. 3.2.3 Emergency function). Thus the needed license file may be ordered with the date of productive use of the PI-CONNECT OPC+ Interface whereby start time of the service contract will also not be until that date.

2. Install und Uninstall

2.1 Requirements for the Installation

To install the OPC+ IF successfully the following requirements are necessary:

- Operating system: depends on the release of PCS7. Please refer to release notes of PCS7 OS clients.
- Operating system login of the interface PC and the PCS7 servers must be identical (also for the PCS7 OS clients).
- Option Archive Recovery (starting with PCS7 Version 6): License for OPC HDA server:
  - Open PCS7 server installed on the interface PC
  - Connectivity Pack assigned to the right WinCC installed on every PCS7 OS server,
  - or since PCS7 V7.1: Open PCS7 for PCS7 OS clients on every PCS7 OS server
- Installation of the PI – SDK (including PI API)
  Version see section *1.3 Performance*
  Hints: refer to section *8 Hints for API Installation and PI Server Configuration*
- PCS7 Server, PI-Server and PI-Interface PC must be time synchronised by the same source.

2.2 Installation of the Interface Software

It is important to read the installation hints in the “readme.txt” file included in the delivered installation files before installation of the OPC+ IF. The file may contain important installation tips.

For installation of the OPC+ IF proceed as follows:

- Call the file setup.exe in the directory SETUP of the delivered software
- Follow the installation hints of the installation procedure.
- Edit the **SiOPC_IF.ini** file according to your needs. Refer to section *3 Configuration*
- The “OPC_Core_Components.msi” is copied to the installation path during installation. If you experience problems on the first start of the interface install the OPC core components.

2.3 Uninstall

To uninstall the OPC+ IF carry out the following steps:

- Terminate all instances of the PI-CONNECT OPC+ interface
- Delete all shortcuts referring to the file **SiOPCIF.exe**.
- Uninstall the software like any other Windows software.

The configuration and log files and the installation directory are not deleted by the uninstall procedure and may be deleted manually afterwards.

3. Configuration

3.1 Structure of the file SiOPC_IF.ini

The configuration of the PCS7 Interface proceeds by setting the parameter in the **SiOPC_IF.ini** file, which has to exist in the Windows NT home directory, normally **C:\WINDOWS**.

The SiOPC_IF.ini file is divided into sections. Each section starts with:

[**XX**].

XX indicates the number of the section. Via this number various instances of the OPC+ IF receive individual parameter sets. Each instance interlinks to one PCS7 server. That means for each PCS7 server one instance of the interface must be configured. The section number always has to be indicated with two digits (e.g. [01] or [12]).

Each section contains 4 blocks of parameters. These are:

- License settings (only for elder versions)
- general parameters
- PI system parameters
- OPC parameters

Parameter may be compulsory, optional or constant.

Compulsory parameter must be indicated. Optional parameters are substituted by standard values if they are not indicated. Constant parameter may not be modified. Some parameters are only read once when starting, others are read cyclic every 30 seconds.

Listing of all types:

Type	Meaning
K	Constant parameter
Op	Optional parameter, only read during start up
OpZ	Optional parameter, read cyclically
Ob	Compulsory parameter, only read during start up
ObZ	Compulsory parameter, read cyclically

For optional parameter the indicated values in all the tables below are the default values which are used if the parameter is not indicated.

3.2 Licence settings

3.2.1 Creating a licence

- The software is licensed to the machine on which it is running. To receive a valid license file proceed as follows
- Copy the CreateUMI.exe and the UMIL.dll (contained in the delivery package) to a directory, e.g. C:\umi
- Run the file CreateUMI.exe
- When you see the "press any key" in the opened window, finish the operation by pressing any key.
- A file named UMI_xxxxxx.dat will be created, where xxxxxx is the name of the PC where you did run the CreateUMI.exe.
- Send this file to us (e.g. info.rhm.industry@siemens.com).
- You will receive the license files for the requested software.
- To have the license files available for installation it might be useful to do this some time in advance to installation.

3.2.2 License Activation

If not already done perform the steps described in *chapter.3.2.1 Creating a licence*

Copy the files UMIL_nnnnnn.dat and UMIL_nnnnnn.txt into the installation path. The files are only valid for the computer for which they have been ordered.

3.2.3 Emergency function

In case of a damage of the hardware the interface software can be installed on another PC. To run the interface on another PC the software contains the following emergency function:

Duration is limited to 14 days. After this time span the software terminates. Within this time span the software can be used without restrictions. This function is also possible for the initial run of the software.

To use the emergency function, the existing licenses have to be installed. Otherwise the delivered, general licenses can be used for the emergency functionality.

3.3 General settings

Within the group of general parameters those settings are located which are not explicitly assigned to the coupling to the PI System or to the coupling to OPC.

Type	Parameter	Explanation
ObZ	Logfile=C:\OPC_IF_01.log	Logfile for output of trace lines should be different for each instance.
Op	FileSize=10000	Maximum size of the log files in Bytes. The file is renamed in *.old when this size is reached. Thereby is assured to have a maximum of two files on the hard disc with the stated size.
OpZ	PrintLevel=5	Trace level for output on the screen 1: major error 2: error 3: warning 4: status variation 5: Point updates summary 6: Point update for each point >7: only for DEBUG purpose, **N O T** for normal use!
OpZ	LogLevel=5	Trace level for output in the log file. Meaning similar to PrintLevel.
*)	Cycle_1=1000,1,1	Scan cycle 1 for cyclic reading of values, refresh flag, DataChange Flag (refer to section *5.3 Configuration of acquisition cycles).*
*)	Cycle_2=1000,1,1	Scan cycle 2 for cyclic reading of values, refresh flag, DataChange Flag (refer to section *5.3 Configuration of acquisition cycles).*
*)	…and so on.	
OpZ	BufferLimit =100000	Maximum amount (number of process values) of the internal buffer between OPC and PI Interface. Range: 1000... 1000000 recommended: 100000
Op	ValueState_Bad=-255	PI Point state, written if the OPC Variable has set the Quality Flag meaning “Bad“. Range: -1024 .. 0 (alle PI States) default: -255 (BAD INPUT)
Op	ValueState_Uncertain=0	PI Pointstatus, written if the OPC Variable has set the Quality Flag meaning “uncertain“. Range: -1024 .. 0 (all PI States) default: 0 (OK, resp.. no State)

*) It is possible to define 30 different intervals for the cyclic reading of values. The intervals must be defined completely. It is not possible to define interval_1 to interval_5 and interval_7 without prior definition of interval_6. At run time no intervals may be deleted or modified. Nevertheless, it is possible to add further intervals (*please refer to chapter 5.3 Configuration of acquisition cycles).*

3.4 PI System Parameters

Listed parameter are applied to configure the coupling to the PI system as well as to control the behaviour of the OPC+ IF interface towards the PI system.

Type	Parameter	Explanation
Ob	PIHost=PI_SERVER	PI Server to be connected
Ob	PointSource=kkkkkkkk	Point Source character to assign the PI point to the interface (max 8 characters, no Upper / Lower case determination
Ob	IORateTag=ior_OPC_IF_1	Tag where a recording rate (values per second, average of ten seconds) of the interface is stored. For each instance a separate tag should be used.
OpZ	UpdateDelay=30	Period in seconds to wait for point updates to be taken in processing by the interface. Range of values : 0..120 seconds Recommended : >= 30 seconds
OpZ	PIAPIVerbose=1	Display use of PI-API 0: no display 1: only errors 2: proceeding of any function Recommended : 1
Op	USESDK= 0	0: Configuration parameters are read using API 1: configuration parameters are read using SDK (slower)

3.5 OPC DA / OPC HDA Parameters

These parameters are used to define the linking to the OPC server of a stand alone PCS7 server or to both OPC server of a redundant PCS7 server pair. Please take in regard especially the parameters mentioned in Chapter "*5.16 Controlling of the OPC connection via an OPC test Item"*.

Type	Parameter	Explanation
Ob	MasterOPCServer = OPCServer.WinCC	ProgID of the first respectively stand alone OPC DA server. For the PCS7/WinCC OPC DA Server the ProgID always is "OPCServer.WinCC".
Op	SlaveOPCServer = OPCServer.WinCC	ProgID of the redundant partner. In case of a stand alone OPC Server this values should be omitted. Please refer to *chapter 5.2 Connecting to redundant PCS7 OS (redundant OPC server).* For the PCS7/WinCC OPC DA Server the ProgID always is "OPCServer.WinCC".
Ob	MasterOPCServerHost= Serv1	Computer name of the PC where the MASTER OPC DA server is installed.
Op	SlaveOPCServerHost= Serv2	Computer name of the PC where the SLAVE OPC DA server is installed. This value only has to be set if a SlaveOPCServer is defined. Also refer to c*hapter 5.2 Connecting to redundant PCS7 OS (redundant OPC server).*
Op	MasterHDAOPCServer = OPCServerHDA.WinCC	ProgID of the first respectively stand alone OPC HDA Servers. For the PCS7/WinCC OPC HDA Server the ProgID always is "OPCServerHDA.WinCC".
Op	SlaveHDAOPCServer = OPCServerHDA.WinCC	ProgID of the redundant partner. In case of a stand alone OPC Server this value should be omitted. Please refer to *chapter 5.2* Connecting to redundant PCS7 OS (redundant OPC server). For the PCS7/WinCC OPC HDA server the ProgID always is "OPCServerHDA.WinCC".
Op	MasterHDAOPCServerHost= Serv1	Computer name of the PC where the MASTER OPC HDA server is installed.
Op	SlaveHDAOPCServerHost= Serv2	Computer name of the PC where the SLAVE OPC DA server is installed. This value only has to be set if a SlaveHDAOPCServer is defined. Please refer to c*hapter 5.2 Connecting to redundant PCS7 OS (redundant OPC server).*

Op	MasterOPCDAItemPrefix, MasterOPCDAItemSuffix, SlaveOPCDAItemPrefix, SlaveOPCDAItemSuffix, MasterOPCHDAItemPrefix, MasterOPCHDAItemSuffix, SlaveOPCHDAItemPrefix und SlaveOPCHDAItemSuffix	Definition of OPC DA and OPC HDA Item prefix und suffix using the according option. Refer to chapter *5.17 Option Connection specific Item Prefix* / Suffix
Op	OPCHDAMaxRequestTime	Time span limit in hours for an OPC HDA request. If a request is done with a greater time span the request is parted in the according number of partial requests. Default is 144 (=6 days)
Op	OPCDATestItem	Item to verify the connection state of the OPC DA server. Refer to chapter *5.16 Controlling of the OPC connection via an OPC test Item*
Op	OPCConnectionTestTime	Cycle of the OPC connection checks. Refer to chapter *5.16 Controlling of the OPC connection via an OPC test Item*
Op	OPCTestItemTSTimeout	Timeout in seconds for the OPCDATestItem. Refer to chapter *5.16 Controlling of the OPC connection via an OPC test Item*
Op	OPCReconnectDelay	Time delay in seconds between two reconnects
Op	OPCDAItemBadStateSupression	Delay in seconds for the OPC DA BAD States of an item to be forwarded to PI. Necessary for Matrikon Tunneler because it marks all items as BAD in case of a network connection error before the interface may react. The time should be set to twice the time of PCConnectionTestTime(default=15) for Matrikon Tunneler. In other cases the function should be switched of with the value of 0 (default)
Op	LCID	LOCALID of the OPC connection. Default is 1031 (Germany)
Op	Archive_1= local::Prozesswertarchiv\	Serially numbered list of the process value archives. Please refer to chapter *6.9 Location5 Attribute (PCS7 / WinCC Archive) and chapter 7.3 Identification of the WinCC Process value archives.)*
Op	Archive_2= local::Prozesswertarchiv2\	Serially numbered list of the process value archives. Please refer to chapter *6.9 Location5 Attribute (PCS7 / WinCC Archive) and chapter 7.3 Identification of the WinCC Process value archives*
Op	HDADataBlock = 750	Maximum size of a data block (number of archive values) to be read in a single request out of the PCS7 archive. Thereby the load of the PCS7 server in HDA mode may be controlled. Small sized values here and big sized values for “HDADelay” result in a lower load of the PCS7 server but will extend the time to stay in HDA mode. Range: 1… 65535 Recommended: 1000
Op	HDADelay=700	Time in ms to wait for the next data block via the HDA OPC server. Higher values will decrease the load on the PCS7 server. Range: 0 ... 65535 Recommended: 100
Op	OPCStringMode=0	Mode used to read or write PI String tags from the OPC Server. 0: Normal mode (default). String is written as VT_BSTR respectively OPC value is transformed to VT_BSTR before further processing 1: read and write string as ByteArray from /to OPC. Interpretation as 8 Bit/character String. 2: like 1, but interpretation as 16 Bit/character String. 3: request a OPC Item as VT_BSTR (otherwise always VT_EMPTY)
Op	WindowRecover=0	If “WindowRecover“ is set to ‚1’, the interface works in HDA Window Recovery mode. Please refer to chapter *5.12 HDA Window Recovery Mode*.
Op	WindowRecoverStartTime= 2006-01-25 16:17:54	Start time of HDA Window Recovery mode. Refer to chapter 5.12 HDA Window Recovery Mod. Format is mandatory : YYYY-MM-DD hh:mm:ss
Op	WindowRecoverEndTime= 2006-01-25 22:17:54	End time of HDA Window Recovery mode s. Refer to chapter 5.12 HDA Window Recovery Mod. Format is mandatory: JJJJ-MM-DD HH:MM:SS
Op	OPCWriteProtection = 1	1: Write protection activated, OPC variables can not be written into the OPC Server 0: Write protection deactivated, OPC variables may be read and written

Op	OPCS7WAMaxReadds=3	OpenPCS7WorkAround; Number of re-add trials of a variable. Refer to Chapter. 5.19, OpenPCS 7 Workaround
Op	OPCS7WAReaddQualityCode=4	OpenPCS7WorkAround; OPC error code. Refer to Chapter. 5.19, OpenPCS 7 Workaround
Op	OPCS7WAReaddDelay=25	OpenPCS7WorkAround; Delay between two re-add trials. Refer to Chapter. 5.19, OpenPCS 7 Workaround

3.6 Global Parameters (for all instances valid)

3.6.1 Replacement tables for digital conditions (DigiCodeTable)

It is possible to define Replacement tables for digital conditions. In these tables, a PI status value is dedicated to a hexadecimal OPC process value. These can be values from a digital Set (>=0) or values from a PI system status (<0). For not having to itemise every single process value, a Default Value can be defined which is used whenever no entry for the OPC process value exists in the table. The replacement table starts with an own section [DigiCodeTable_xxx]. “xxx” stands for the number of the table. Tables from 001 till 999 can be defined. When you enter the number in PI Tag in “UserInt1”, the tables are referenced. Therefore see *Chapter 5.8 Replacement tables for digital conditions.* Example for a replacement table:

```
[DigiCodeTable_001]
Code_def=-255
Code_001=0x00, 4
Code_002=0x01, 3
Code_003=0x02, 2
Code_004=0x03, 1
Code_005=0x04, 0
```

4. Starting and terminating of the Interface

4.1 Starting as Windows console application

Hint: Whenever the archive recovery option of the interface is used make sure that the archive comparison of a redundant PCS7 server pair has been finished completely before starting the interface. If this is omitted gaps in the PI point data may appear which can no more be closed by the interface. Closing of PCS7 archive comparison is indicated by a corresponding message in the alarm system of PCS7.

Start OPC+ IF as Windows Console application by inserting (e.g.)

\\Programme\Siemens\SiOPC \SiOPCIF <X>

The command line parameter <X> equals the interface's instance number. This parameter controls the access to the various sections of the SiOPC_IF.ini file. The instance number determines to which PCS7 servers a coupling is established. Calls may look as follows:

C:\Programme\Siemens\SiOPC \SiOPCIF 01 for instance 1
C:\Programme\Siemens\SiOPC \SiOPCIF 03 for Instance 3

Complete the parameter file **SiOPC_IF.ini** by a section for the new interface instance and edit the required parameters.

Please notice:

- Each instance requires a separate license.
- Instance numbers 01 to 99 are tolerated for normal mode.
- Instance numbers above 100 are only be used for funktion „Online Window Recovery" (see section *5.12 HDA Window Recovery Mode*)

4.2 Simplified start as Windows console application by linkages

To start several instances it is convenient to be able to start the instances via desktop symbols. Please proceed as follows:

- On the Windows desktop for each requested instance create a link referring to the file **\\Programme\Siemens\SiOPC \SiOPCIF.exe**. Use reasonable indications for the connections, e.g. indication of PCS7 server coupling the instance (<PCS7ServerName>_to_PI).

- With a right mouse click activate the connection's dialogue "Features", select table "connections" and insert the required command line parameter (instance number) in field **"target"** behind **SiOPCIF.exe**.

  e.g.: "**C:\Programme\Siemens\SiOPC \SiOPCIF" 01** **for Instance 1**
  "**C:\Programme\Siemens\SiOPC \SiOPCIF" 03** **for Instance 3**

- Activate the connection's dialogue "features", select table connections and select option **"minimized"** in field **"execute".**

Complete the parameter file **SiOPC_IF.ini** by a section for the new interface instance and the required parameter. Please notice:

Please notice:

- Each instance requires a separate license.
- Instance numbers 01 to 99 are tolerated for normal mode.
- Instance numbers above 100 are only be used for funktion „Online Window Recovery" (see section *5.12 HDA Window Recovery Mode*)

4.3 Automatic start up as console application after Login

Setting up the automatic start up of the interface software requires three steps:

1. Set up of the Windows Auto Login

First the operating system has to be set to start up with the “Autologin”. This means a user is logged in automatically without any interaction. As a consequence programs located in “Start→Program Files→Start up” will be executed. Login and Password of this user must be identical with the operating system user of the PCS7 server.

To set up the Autologin contact your system administrator or make use of the tool “LockWorkStation” included in the interface software delivery. Starting with Windows 2000 the workstation may be locked directly after the login with this tool.

2. Creating a Batch-File

For automatic start up of the interface instances after the Autologin a batch file (e.g. named Start.bat) has to be created. The Windows text editor (e.g. notepad) may be used for this. The file should have the following structure:

```
@echo off
echo Start job of die SiOPC-Interfaces

rem wait 1 minute
call wait 60000

rem start the 1st interface Instance
start C:\Programme\Siemens\SiOPC\SiOPCIF.exe  01

rem for 2nd instance only

rem wait 30 sec.
rem call wait 30000

rem start 2nd interface Instance
rem start C:\Programme\Siemens\SiOPC\SiOPCIF.exe  02
```

Start.bat file for automatic start up after system start up

Adjust paths and further instances on your requirements.

To use the command “wait” requires copying the file “wait.exe” to the system directory (e.g. c:\windows). The file is included in the delivered interface software

3. Activate Auto start

Copy the batch-file to the start up directory of the auto login user. The path might look like:

„C:\WINNT\Profiles\Administrator\Startmenue\Program Files\start up\“

With the next start up of the operating system the instances of the SiOPC+ Interfaces will be started automatically.

4.4 Shutdown of the console application

To shutdown the SiOPC+ IF proceed as follows:

- Activate the console application window
- Press **<Strg + C>**, to shutdown the SiOPC+ IF.
- The interface will terminate. Please wait until the window of the console windows is closed.

4.5 Starting as Windows Service

4.5.1 General

PI Connect OPC+ Interface (V3.8 and higher) can be started as a Windows service. The necessary steps are described below. Please consider always the following details:

- If the interface (=OPC client) shall access an OPC server remotely, it has to be ensured that the user running the service has accordant DCOM access rights to the OPC server (and also backwards for the callbacks of the asynchronous OPC interface). This is achieved normally by using a user account for the service which is available on the OPC server, too.

- Some OPC servers (e.g. Open PCS7) are only released to operate on a PC using a so-called „Autologon“. Therefore on such a PC a user is always logged in automatically. This has to persist if the interface will run as a service, too!

4.5.2 Register, configure and operation of an Instance as a service

For registration of the instances # 1 – 5 as a service, the interface software creates shortcuts in the installation path of the interface (e.g. C:\Program Files (x86)\Siemens\PI CONNECT OPCPlus (Stand alone)). However, if instances with numbers > 5 shall be registered the following command can be used:

SiOPCIF -i nn

(where nn = Instance number 01 to 99).

After registration the following service can be found in the windows service manager:

Display name: *PI CONNECT OPC+ Interface (Standalone) # nn*

Service name: *PI_CONNECT_OPCPlus_Interface_SA#nn*

Nn means the instance number, too. Thereafter the service may be configured using within the Windows services. Usually the following parameters are set:

- User account for the service (refer to chapter 4.5.1 General)
- Startup type (manually or automatically)
- Recovery

For the last item we recommend to use the windows option which restarts the service after failure.

User account and startup type to be used depend on the intended use. Added two recommended examples:

Configuration 1: local connection to an Open PCS 7 OPC Server

- User Account corresponds with the Windows account of the PCS7 client stations. This is often identical with the Windows account of the OS Server.
- Startup type: manual (look below)
- Recovery: restart the service

Running in combination with Open PCS7 an autologon must be performed before starting the Open PCS7 OPC Server. This has to be configured accordingly. The interface software must not be started before the logon. Startup of the interface software should be done with a Net Start Command included in a Batch file in the Startup directory.

Autostart.bat

```
echo Startjob for SiOPC-Interfaces

net  start  PI_CONNECT_OPCPlus_Interface_SA#nn
```

Configuration 2: remote connection to WinCC OPC Server (OS Server)

- User account corresponds to the Windows account of the OS Servers.
- Startup type : Automatic
- Recovery: Restart the service

Therefore the PC does not need an Autologon. The interface may be operated without a logged in user.

4.5.3 Removing service of an instance

If an instance is no more needed the service may be unregistered. The instance must be stopped in advance! For the first 5 instances a shortcut for unregistering is created in the installation path of the interface software (e.g.: C:\Program Files (x86)\Siemens\PI CONNECT OPCPlus). For all further instance the following command line may be used to unregister the service:

SiOPCIF -u nn (nn = instance number 01 to 99).

After unregistering the instance is no more available in the Windows services.

5. Function of the Siemens OPC+ Interface

5.1 Scope

Reading process values:

Process values are read cyclic out of the Data Manager (internal and external variables) of a PCS7 server in up to 30 free definable cycles. However, the cycles are adjusted to the cycles of the OPC server (please refer to chapter 5.3 Configuration of acquisition cycles).

In the online mode (using OPC DA mode), the process data transfer between the DCS and the PI interface will run only on data change. Due to that the DCS has a lower load. For cyclic supply of the PI exception test (check for horizontal value changes) events with actualized time stamps are built (*see section 5.4 Special feature on time stamping*).

Either the stamps out of the process control system or local time if the Interface PC may be used.

Writing process values:

Data out of the PI System may be written to PCS7 variables in the Data Manager. Writing is done event based (OnDataChange) whenever the source point of the PI system delivers a new event.

Archive recovery via OPC HDA (optional)

For PI points using the function the data gap since the last run of the interface is closed by the interface software. To do so the values are read out of the process value archives of the PCS7 /WinCC servers via the OPC HDA Server.

"Archive Recovery" is an option of the interface software and must be licensed separately to use the option.

Connection check:

The connection to the PCS7 server and the PI System is permanently checked by the SiOPC+ interface and a reconstitution of the connection is established. In case of loss of connection to the PCS7 Server data gaps occur. In case of breakdown of the connection to the PI System data are completely transmitted after successful reconnection. For this activation of the API Node Buffering for the SiOPC+ interface is required *(refer to chapter 8, Hints for API Installation and PI Server Configuration*).

5.2 Connecting to redundant PCS7 OS (redundant OPC server)

Shall the PI-Connect OPC+ interface be connected to a redundant OPC server pair without using of OpenPCS7 and the handling of the PCS7 redundancy shall be done by the PI-Connect OPC+ interface this is done by configuring two OPC server connections in the file SiOPC.ini (refer to *chapter 3.5, OPC DA / OPC HDA Parameters*). The behaviour of the interface software connecting to a redundant OPC server is defined as follows:

Start up mode:

On start up the interface tries to open a connection to one of the two OPC servers. In case this is not successful the interface tries to connect to the second OPC server. If this is also not possible alternately the connection is tried to establish with one of the OPC servers until a connection is established successfully.

Connection breakdown during runtime:

After a successful start up the interface software is connected to one of two OPC servers of a redundant pair. If this connection is lost the interface switches into start up mode until a connection to one of the OPC servers is established.

A connection loss appears whenever the connected OPC server is shut down or the network connection is interrupted.

The state of some particular process values of the OPC server does not cause a connection loss. In consequence error states on a part of the process values do not cause the interface to switch to the other OPC server.

The interface software activates only one connection to one of the OPC servers. In case of connection breakdown the connection to the redundant OPC server will be established. This may last up to a few minutes depending on the number of variables to be registered. During this time no values are read!

5.3 Configuration of acquisition cycles

It is recommended using only acquisition cycles, what are supported by the OPC Server and his basic DCS. Using of not supported cycles, unwanted effects may occur. Using the example of PCS7, below the recommended configuration of the acquisition cycles is described.

Because the intervals stated here are mapped to the cycle times in PCS7 it is useful to use its cycles here, too. Cycle times less than 1 second are not expedient. In accordance to this result the following recommended cycles (PCS7 standard cycles):

Cycel_1	=	1000	*{optionally}* ,1 ,0
Cycel_1	=	1000	*{optionally}* ,1 ,0
Cycel_3	=	1000	*{optionally}* ,1 ,0
Cycel_4	=	2000	*{optionally}* ,1 ,0
Cycel_5	=	5000	*{optionally}* ,1 ,0
Cycel_6	=	10000	*{optionally}* ,1 ,0
Cycel_7	=	60000	*{optionally}* ,1 ,0
Cycel_8	=	300000	*{optionally}* ,1 ,0
Cycel_9	=	600000	*{optionally}* ,1 ,0
Cycel_10	=	3600000	*{optionally}* ,1 ,0
Cycel_11	=	2000	*{optionally}* ,1 ,0
Cycel_12	=	3000	*{optionally}* ,1 ,0
Cycel_13	=	4000	*{optionally}* ,1 ,0
Cycel_14	=	5000	*{optionally}* ,1 ,0
Cycel_15	=	10000	*{optionally}* ,1 ,0

Cycle times are in milliseconds. Thereafter the switches for the kind of data communication between the OPC server and the PI-CONNECT OPC+ Interface may follow comma separated. The first switch sets the cyclic “refresh” to on; second one sets the “DataChange” notice.

When “DataChange” is activated (=’1’) all actual changes of the process values will be written in the internal cache and also directly forwarded to the PI API.

When “Refresh” is activated (=’1’), the interface tests cyclically the cache and forwards the last received value to the PI API. Those PI tags who are configured for a cycle, that is activated for Refresh, in each case the IO Type 1 or 3 should be used (refer also section *6.7 Location3 Attribute (IO-Type)* and *5.4 Special feature on time stamping*).

For optimal mode of operation both switches (Refresh and DataChange) should be set to ‘1’. **This is also the default value if those switches are not present.** In this case the interface forwards all actual changes of process values reported by the OPC server (Data Changes) directly to the

PI-API and in case of an constant development without changing of the process value, the PI-API will be supplied cyclically with the last received process value from the interface cache but with an adapted time stamp (only with IO Typ = 1 or 3).

In each case the transmission between OPC server and the interface is done only, if there is a change of the process value (On Data Change). The interface ensures itself the cyclical feeding of the PI API using his internal cache. Thus the load for the OPC server and the subjacent process control system (DCS) will be minimized.

Cycles 1 to 10 are PCS7 standard cycles and can not be changed in PCS7. Cycles 11 to 15 are user specific cycles which may vary. They should be adjusted in the file SiOPC_IF.ini according to the actual settings in the PCS7 server.

Process values may be read cyclically from a PCS7 server in 30 free adjustable cycles. These cycles have to be defined in the section of the SiOPC-IF.ini file belonging to the interface instance (please refer to *chapter 3 Configuration*). These times specify the cycle to order the values from the OPC server. The WinCC OPC server maps these requests to the 15 cycles of the PCS7/WinCC server. Cycles not contained in the list of the 15 PCS7 server cycles are mapped to the next faster cycle.

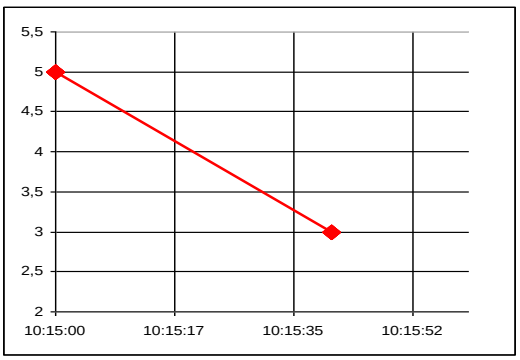
All process values having the same cycle are signed up for one group in the OPC server. A definite offset for the start of the scan cycle is not adjustable. Processing starts in groups with a delay of some seconds after reading the configuration out of the PI server.

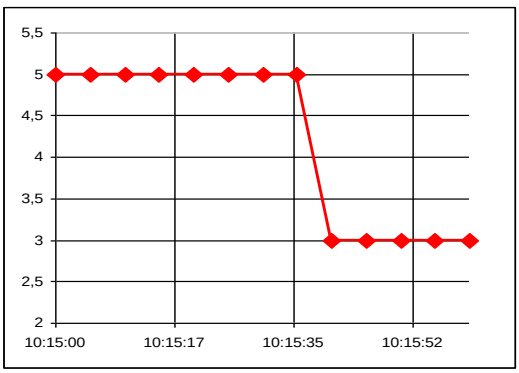
Directly after the start of data collection for a group the process values are read. Because of this on start up of the interface the first values for all cyclic reading points will appear in the PI system after a short time.

An optimisation algorithm assures to un-register no more needed process values (caused by a PI Point update “DELETE”). However, on very large changes in configuration we recommend the interface be restarted.

5.4 Special feature on time stamping

The PI-Connect OPC+ Interface works in the face of the OPC Server of the DCS in a way to generate a minimum of load on the DCS. Thus only value changes are transmitted from the OPC server to the PI-Connect OPC+ Interface. If a value is constant for several reading cycles no new value will be transmitted to the interface. If only these changed values are provided to the PI API, reproducing problems would appear in the PI archive. Especially values running for a long time constantly with a sudden jump appearing will not be shown as a jump but as a ramp (left picture).





To avoid those reproducing problems in the PI archive the PI-Connect OPC+ Interface does some corrections. The PI-Connect OPC+ Interface itself provides cyclic values in the scan cycle of the PI API and delivers a corrected (actual) time stamp if needed. Leaps in time resulting of PCS7

server redundancy switches have to be taken in account in doing so. To avoid faulty values caused by cycle imprecision it may be necessary to suppress the value for one scan cycle for measure points which run constantly for at least two scan cycles.

Therefore it is an absolute must to have the PCS7 Server, the Interface PC und the PI Server synchronic in system time. If this can not be assured we recommend to use the local time of the Interface PC for time stamping of the values sent to the PI System.

In the following example is shown how the value changes received from PCS7 are forwarded cyclically to the Pi API and how the first level compression of PI System (exception test) works:

Scancylce = 1s; ExMax = 600; ExDev = 0,5

Time stamp	New Value out of PCS7	Sent to the API by the interface	Value sent by the PI-API to the PI Server Snapshot table (uses time stamp of column)
00:00:00	3	3	3
00:00:01		3	
00:00:02		3	
00:00:03		3	
00:00:04		3	
00:00:05		3	3
00:00:06	5	5	5
00:00:07		5	
00:00:08		5	5
00:00:09	8	8	8
00:00:10		8	
00:00:11		8	
00:00:12		8	8
00:00:13	2	2	2
00:00:14		2	
00:00:15		2	
00:00:16		2	

5.5 Writing PI data “OnDataChange”

Usually it is required to configure a point as a reference. This reference-point possesses the PointSource of the OPC+ IF and includes the required parameter for the connection to the PCS7 server. The attribute “source tag” of this reference point makes contact to the source point whose data are transmitted to the PCS7 server. All values transmitted from the source point to the PCS7 server are stored in this reference point. The reference point contains a copy of all the source point’s data which were sent to PCS7.

If it possible to assign the PCS7 interface’s PointSource to a source point and to store all required parameters in its attributes the reference to another source point is not needed. The attribute “SourceTag” has to be assigned to the tag itself in this case. Copying of the sent data is avoided in this case.

5.6 Bit masking

It is possible to select a single bit or a group of bits out of the value using a 32 bit mask (refer to *chapter 6 PI Point configuration)*. For this the PCS7 source variable is converted into a 32 bit unsigned integer. Post decimal positions of float variables are truncated (not rounded!!). If the value of the source point is out of range of an unsigned 32 bit integer the PI value is set to "OVERRANGE" res. "UNDERRANGE". The bit mask may be given binary (max. 32 digits), hexadecimal (max. 8 digits) or decimal (0....4294967295) separated by a blank following the PCS7 Variable name in the ExDesc. The keyword "BITMASK" and the format key "0x" res. "0b" are case sensitive. The first character of "0x" and "0b" is the numeral null!

Examples:

BITMASK=0b1100:	Binary bit mask. Bit 2^2 and2^3 will be selected. Only the decimal values 0, 4, 8, 12 may be written to PI.
BITMASK=0xf0:	Hexadecimal bit mask. The bits 2^4 to 2^7 will be selected and only the decimal values 0, 16, 32, 48, 64, 70, 86, 102, 118, 134, 150, 166, 182, 198, 214 und 230 may be written to PI.
BITMASK=10:	Decimal bit mask. The bits 2^1 and 2^3 will be selected. The following values may be written to PI: 0, 2, 8 and 10.

Optionally it is possible to do a right shift on the masked double word to set the first selected bit to the position 2^0. This is helpful if e.g. the bits 2^6 and 2^7 are selected (BITMASK=0b11000000).Without the shift option the values 0, 64, 128, 196 may be written to PI. With the shift option (BITMASK=0b11000000; SHIFT=1) it will be shifted 6 positions to the right because 2^6 is the lowest with the mask selected bit. Therefore the bits 2^6 and 2^7 will turn to 2^0 and 2^1 and with this the gapless with zero starting values 0, 1, 2, 3 will be written to PI. This is a relevant help in designing digital sets within PI. The shift option is activated in adding the keyword "SHIFT=1" separated by a semicolon to the bit mask.

5.7 Scaling

The OPC+ IF supports scaling of values received from the OPC server. The function is activated by setting the location2 attribute of the PI tag to "1". The attribute UserReal1 contains the value of the factor; in UserReal2 the value of the offset is configured. Please refer to *chapter 6 PI Point configuration*. The value transmitted to the PI server is calculated as follows:

Value PI = (OPC value * UserReal1) + UserReal2

Value PI = (OPC value * factor) + offset

5.8 Replacement tables for digital conditions

To alleviate the data acquisition for PI Points of type "digital", replacement table can be defined in the configuration file(see *Chapter 3.6.1 Replacement tables for digital conditions (DigiCodeTable)*.In these tables a definite number of OPC process values can be defined to PI status values (<0: system, otherwise Digital Set). All not itemised process values are defined with the value default.

To net this function for a point, the replacement table has to be defined in the configuration file before creating the PI Point. At create the PI tags the number of the replacement table is deposited in "UserInt1". If a replacement table is not required, the value '0' has to be placed there.

5.9 Archive Recovery OPC HDA (optional)

5.9.1 General and requisites

The option “Archive Recovery” of the OPC+ IF is able to use this way (HDA) to close the gap in the data since the last shutdown of interface on start up (e.g. after service works on the interface PC or the network).

Prerequisites to use the archive recovery function are:

- PCS7 Version V6.0 or higher, with licence for OPC HDA server
- Siemens PI-Connect OPC+ Interface with the license for option archive recovery (HDA)
- The tags must be defined in the PCS7 process value archive.

5.9.2 Functionality

Starting with PCS7 version V6.0 the PCS7 Server, precisely WinCC supports an OPC HDA server besides the OPC DA server. Via the OPC HDA server the process value archive of the PCS7 server may be accessed. The option “Archive Recovery” of the OPC+ IF may use this way to close the data gap since the last shutdown on start up (e.g. after service works on the interface PC or the network). The interface detects the time stamp of the last value in the PI archive and asks for values out of the PCS7 system starting with that time. When the actual system time is reached data collection is switched over to OPD DA. Because OPC DA is signed up in an early stage the switching gap will be < 5 seconds. The archive recovery function may be activated for each PI tag separately.

For each PI tag that uses the archive recovery function, the according PCS7 variable must be available in a PCS7 process value archive. A breach of this condition in a huge amount may cause errors in the interface software.

The archive variable name must be specified because normally variables in the process value archive are different from the variable names in the DataManager. Therefore the “intrumenttag” attribute of the PI tag is used. Furthermore the archive containing the variable must be specified. For this the attribute “location5” is foreseen. Points not using this functionality should be set to „0“. For further information please refer to *chapter 6 PI Point configuration*.

Archive recovery makes sense only for PCS7 archive variables corresponding with the chronological sequence of the process value. In case of having archived only calculated values e.g. the total of the last ten values archive recovery does not make sense because in the example you only have a value archived which is about tenth the value.

5.9.3 Performance

Performance of the archive recovery function depends on the starting point. The farther the starting point lies in the past the less the performance will be. Here is an example of a performance test:

Starting point:	**Performance:**
4 days in the past	180 archive values per second
1 day in the past	800 archive values per second
1 hour in the past	1400 archive values per second

These values also depend on the load of the PCS7 server. The actual performance is shown in the interface application window during archive recovery. Be aware that processing is done separately for each tag. Once a tag reaches the actual time this tag is switched to online mode and the archive recovery process starts with the next tag.

5.9.4 Archive recovery system load

Whenever a large data gap (> 1 day) is to be filled for a high amount of PI tags (>100) you may reckon a higher load of the PI server and the PCS7 server, particularly when the PCS7 archiving is done in shorter cycles (<10 sec). Please keep in mind:

The PI archive must have enough storage space left

Example: Archive A starts at 01.Apr.2004 04:55:30am. At 10:30:00 am the interface is shut down. Assumed at this time the archive is filled up to 80%. Also assumed ten hours later the archive is filled up to 95% and PI system shifts to the next archive. Shortly after the interface is restarted. Archive recovery starts with the time range 10:30am to 8:30 pm (10h gap). These 10 hours have to be filled into the archive which has 95% of its capacity used. If this is a fixed archive an overflow may occur and data will be lost. Because of this it is very important to force an archive shift if needed after stopping the interface. In case of a very long shutdown it might be useful to force further archive shifts. However (may be the interface crashed) before starting the interface the free amount of archive space should be checked. Eventually fixed archives should be transferred to dynamic archives to handle that.

The PCS7 servers have to be in stable state.

On a redundant PCS7 Server pair archive recovery should only be done with both servers working and their own archives being synchronized. The load on the PCS7 servers should be looked at.

5.10 Supported Data Types and their conversion

5.10.1 List of supported Data Types

The following PI Data Types are supported:

1. Digital
2. Int16
3. Int32
4. Float16
5. Float32
6. Float64
7. String

Not supported is the PI Data Type:

- Blob

The following PCS7 Data Types are supported:

a) Binary Variable
b) Unsigned 8-Bit Value
c) Signed 8-Bit Value
d) Unsigned 16-Bit Value
e) Signed 16-Bit Value
f) Unsigned 32-Bit Value
g) Signed 32-Bit Value
h) Float 32-Bit
i) Float 64-Bit
j) String 8-Bit Character set
k) String 16-Bit Character set

The following PCS7 Data types are not supported:

- Raw data
- Text reference
- All structured data types

5.10.2 Data Type Conversion for reading [1] Points

PCS7 numeric (a to i) → PI numeric (2 to 6):

By converting the numeric PCS7-Types (a to i) to numeric PI-Types (2 to 6) loss of exactness may appear depending on the destination tag data type whenever the source data type has a higher resolution than the destination tag data type . If the value will be out of range of the PI Types the PI - Points get the state „OVERRANGE“ or „UNDERRANGE“.

[1] Read PCS7 values and write to PI

PCS7 alphanumeric (j and k) → PI numeric (2 to 6):

Conversion of PCS7-Types (j and k) to numeric PI-Types (2 to 6) tries to change the text to a numeric value. The decimal delimiter is taken from the country settings of the PC. If the alphanumeric PCS7-Wert does not contain a number, the result is undefined.

PCS7 numeric (a to i) → PI alphanumeric (7):

The numeric PCS7 values are converted character by character to the PI-Type String. The decimal delimiter is taken from the country settings of the PC.

PCS7 alphanumeric (j and k) → PI alphanumeric (7):

Transmission of PCS7-Types (j and k) to the PI-Type "String" are done character by character without any conversion.

PCS7 numeric (j and k) → PI digital (1):

The post decimal positions of the numeric PCS7 value will be truncated (no rounding!) to an Integer value. If this values is in the range of 0 .. 255 it will be transmitted as Digital Code to PI. Values out of the range 0 ... 255 will get the state „OVERRANGE" or UNDERRANGE".

PCS7 alphanumeric (j und k) → PI digital (1):

First the conversion "PCS7 alphanumeric → PI numeric" is done. After this the conversion "PCS7 numeric → PI digital" is done.

5.10.3 Data Type Conversion of writing[2] points

Conversion of writing points is done according to the rules for reading Points with the exception according chapter *5.10.4 Writing to OPC Variables of Data Type VT_BOOL* .

5.10.4 Writing to OPC Variables of Data Type VT_BOOL

OPC Bool variables are mapped to 0,1 in PI, not to 0,-1. For easier writing of Bool Variables to the OPC server the following conventions are implemented:

Writing Bit to OPC server: PI Int16, Int32, digital are written bit masked as VT_I4 if the ExDesc contains: BIT= 0 ... BIT= 15

Writing Bit to OPC server: PI Int16, Int32, digital are written bit masked as VT_Bool if the ExDesc contains: BIT=-0 ... BIT=-15

To write a value stored in PI as 0/1 in a PI digital, Int16 or Int32 to an OPC variable of type VT_BOOL the ExDesc entry must be as follows:

ItemID=<OPC Item ID>;BIT=-1

[2] Read PI values and write to PCS7

5.11 PI Point States used by the interface

Instead of a value a PI Tag may show a state containing additional information to a missing value.

The following PCS7 Errors have associated PI States for the affected Point:

Possible reason	PI state
PCS7 Server not within reach, PCS7 Variable not found, breakdown of connection to PCS7 server, AS not within reach of PCS7 server	Bad Quality
Error by reading the PI Point configuration or no access rights to PCS7 variable (@PCS7 configuration)	Error
Value exceeds range of PI Pointtypes	Over Range
Value falls below range of PI Pointtypes	Under Range
All other faults	Bad

The following states of the interface result in the shown PI States for all Points of the interfaces:

Possible reason	PI state
Interface start up	Scan On
Interface shut down	Scan Off
Loss of the connection to OPC server	Scan Timeout
Re-establishment of the connection to OPC server	Scan On

The interface has the option to define in the ini File the correlation between the OPC quality flags and the PI Point states resulting thereof. The following quality flags from OPC are supported:

OPC Quality Flag	PI Status
Bad (quality code 00xxxxxx)	If the ini-File parameter “ValueState_Bad” is not set the state„-255“ (BAD VALUE) is sent, else the PI state according to the value of the ini-file parameter.
Uncertain (quality code 01xxxxxx)	If the ini-File parameter “ValueState_Uncertain” is not set the state„0“ (OK) is sent, else the PI state according to the value of the ini-file parameter.
Good (quality code 11xxxxxx)	Always „0“ (OK)

The default values ValueState_Bad=’-255’ and ValueState_Uncertain=’0’ match to the behaviour of elder versions.

5.12 HDA Window Recovery Mode

Precondition for HDA Window Recovery si a license for the option „Archive Recovery OPC HDA“. The interface supports 2 modes of HDA Window-Recovery:

Offline:

Window Recovery is done instead of collecting acutal prozess data. The according instance of the interace must be stopped.The resulting actual gap in the PI Archive during Window Recovery will be closed by the standard Archive Recovery after the Window Recovery has finished (refer to chapter. 5.8). In the configuration filef the section fort he according instance must be configured before start of Window Recovery. Therfore the parameters Window Recovery=1 and Start- and End time must be set (refer to chapter *3.5 OPC DA / OPC HDA Parameter)*.

After restart for all PI tags conifgured for HDA Archive Recovery the values fort he defined time range will be read from the HDA OPC Server and send to the PI Server. Thereafter operation stops. There is no automatic switch to DA mode. No actual values for the tags will be read. PI tags without OPC HDA are not scanned. To return to normal operation Windows Recovery mode must be turned off (Window Recovery=0) and the interface instance must be restarted.

Online:

Window Recovery is done in parallel to the scanning of actual process data. In the configuration file the according section fort he instance must be duplicated. Therefore the instance number has to be heightened for 100 of the running online instance.

For Excample for the instance 01 additionaly the instance 101 must be defined and used for Window Recovery verwendet werden. All paramters of instance 1 must be copied for instance 101. The parameters for Logfile name und Window Recovery of the Instanz 101 mus be adjusted. Thereafter the additional instance with the number 101 may be started. This instance uses the license and the PI tags (location1) of the instance 1.

5.13 Logfile limitation

The size of the interface logifle is limited to 10 MB. Thereafter the actual logfile is renamed like follows:

xxxxx.yyy --> xxxxx_old.yyy

An existing fie with the same name will be overwritten.

5.14 Autostop Function

Using this function the interface software may be shut down by an external application. Therefore the interface tries to open the file <Applicationpath>\stop.txt every 10 second. If this file is created by another application the interface succeeds. This is interpreted as signal to terminate. The file is not deleted by the interface!!

For automatic termination the file should be created for about 20 sec and thereafter be deleted by an external application.

5.15 Operation on missing of OPC Server call-backs

The missing of OPC DA Server call-backs is reported and optional a reconnect by the interface can be initiated.

Therefore the INI parameters “OPC_NO_CALLBACK_WARNING_DELAY“ and “OPC_NO_CALLBACK_DISCONNECT_DELAY“ may be set.

The value '0' is default for "OPC_NO_CALLBACK_WARNING_DELAY" and means 'off'. The value '600' is default for “OPC_NO_CALLBACK_DISCONNECT_DELAY“ and means warning after 10 minutes without call-backs.

5.16 Controlling of the OPC connection via an OPC test Item

In the INI file the parameter “OPCDATestItem“ may be defined. This item will be registered at the OPC server. Scancycle for this item is set with the parameter “OPCConnectionTestTime”. If no new time stamp for this item for more than “OPCTestItemTSTimeout“ seconds (INI parameter) is recorded this is treated as a connection loss and a reconnect to the OPC server is initiated. Reconnects are done in the cycle of “OPCReconnectDelay“ seconds.

A special case is the observation of the OPEN PCS7 OPC Server using HDA. To be able to recognize a connection loss between the PCS7 server and the OPEN PCS7 server within the interface software and to switch it to HDA mode the OPCDATestItem must be defined. Because of the variable @RM_MASTER exists in all redundant PCS7 systems the following settings are recommended:

OPCDATestItem	= @RM_Master
OPCConnectionTestTime	= 30
OPCReconnectionDelay	= 15
OPCTestitemTSTimeout	= 0 (mandatory)

The variable @RM_Master changes very seldom and provides no new time stamp most of the times when read. The value 0 for the parameter „OPCTestitemTSTimeout“ changes the observation mode of the variable „OPCDATestItem“ from „on change“ to “on error” and allows the interface to switch to HDA mode on connection return.

5.17 Option Connection specific Item Prefix / Suffix

In praxis the following requirement fort he OPC+ Interface appeared:

A redundant DCS (redundant=double) has an OPC Server running on a separate PC as OPC interface holding the connection to the systems containing the data of the DCS. The redundant data source has normally an identical set of variables. The OPC server does not cover redundancy but shows both variable sets. To be able to address these separately the OPC server uses a specific prefix and/or a suffix.

The OPC+ interface may be configured to access such a system. In the PI tag configuration only the identical part of the OPC Item name will be configured. The connection specific pre- and suffix is set in the configuration file separately for the OPC DA/HDA Master and Standby connection.

To be able to switch between master and standby connection the OPC+ interface it must be able to detect if the systems “behind” the OPC server are able to deliver data. Therefore a so-called “OPC Testitem“ (refer to the previous chapter) must be defined.

5.18 Performance Tags

It is possible to define several performace tags in the section „PI Parameter“ in the ini file. All these parameters are optional. The according PI tag is assigned only via the tag name (like the IORateTag does). Pointsource as well as location1 is not relevant for the interface.

The pointtype is mandatory Int32. Only the “RecoveryStateTag“ may be of type “Digital” alternatively.

Values are written every second on change. If the value does not change the values are written every 60 seconds. PI Exception test is not performed (putsnapshot function). Compression is done.

On startup of the interface the performance tags are set to “ScanOn”.At shutdown they are set to “ScanOFF”.

Tag list:

Parameter name	**Function**
RecoveryStateTag	Operating state of archive recovery 0: OPC connection not (yet) active 1: OPC connection active, HDA active 2: OPC connection active, HDA finished
TC_DA_Tags	Number of tags configured for DA only
TC_DA_HDA_Tags	Number of tags configured for DA and HDA.
ErrorTC_TagConfig	Number of tags with errors on the PI configuration determined by the interface. E.g.: missing ExDesc, HDA tag without entry in the instrumenttag, scan cycle not defined etc. Faulty DA or HDA names will be recognized with OPC notification and are covered by separate performance tags. Info: The sum of TC_DA_Tags + TC_DA_HDA_Tags + ErrorTC_TagConfig is the number of all tags configured for this instance of the interface.
ErrorTC_DAError4DATags	Number of tags configured for DA only and the DA notification is not successful. Normally this is because of the ItemID is not available in the OPC server.
ErrorTC_DAError4HDADATags	Number of tags configured for HDA/DA and the DA notification is not successful. Normally this is because of the ItemID is not available in the OPC server.
ErrorTC_HDAError	Number of tags configured for HDA/DA and the HDA notification is not successful. Normally this is because of the ItemID is not available in the OPC HDA server.
TC_HDA_InWork	The value of this performace counter firstly is the same as the „TC_DA_HDA_Tags“ after successful OPC connection, means all errorless configured HDA tags. During runtime the value runs down to zero. As long as it is not zero the “RecoveryStateTag” shows the value “1: OPC connection active, HDA active at the time”. When the HDA access is finished the “RecoveryStateTag” changes to “2: OPC connection active, HDA finished”. To determine that all tags configured for HDA have been working correctly the performance tags “ErrorTC_DAError4HDADATags“ and “ErrorTC_HDAError“ must be checked in addition for the value “0”.

5.19 OpenPCS 7 Workaround

Running OPEN PCS7 the use of the parameters described in the following may be helpful in some special cases. In case of problems with the connection to Open PCS7 we strongly recommend to activate these parameters. Changes on the standard settings should only be done upon advice by us. The parameters are:

OPCS7WAMaxReadds	= 3	(number of Re-add trials)
OPCS7WAReaddQualityCode	= 4	(OPC error code)
OPCS7WAReaddDelay	= 25	(Re add Delay in seconds)

6. PI Point configuration

6.1 Automatic configuration tool (option)

For this interface an automatic configuration tool is available. A user defined filter is set up on the PCS7 module level. Thereafter the PI tag configuration is created automatically. On any change if the PCS7 configuration the necessary changes for the PI configuration may be created in so called “delta” runs. For further information please refer to the documentation of the configuration tool.

Only structure tags are recognized by the confiration tool. The means that the variables have to be parameters if a WinCC structure type. This is the fact for all PCS7 sturcture tags normally.

6.2 Siemens OPC+ Interface specific attributes

PI Point Configuration is done using the PI System Management Tool (PI-SMT, Add-In for MS EXCEL). Only the specific features of the PI-CONNECT OPC+ Interface to addressing and conversion of the variables are mentioned here:

- PointSource — Assignment to PCS7 Interface
- ExDesc — PCS7 Tag
- Location1 — Assignment to instance
- Location2 — Scaling on/off
- Location3 — IO Type
- Location4 — Scan Class (defined in **SiOPC_IF.ini**)
- Location5 — Number of PCS7 archive (defined in **SiOPC_IF.ini**)
- InstrumentTag — PCS7 archive variable name
- UserInt1 — replacement table
- UserReal1 — scaling, factor
- UserReal2 — scaling, offset
- SourceTag — Source tag for Output Points

This documentation only covers the interface specific attributes. For Pi point configuration additional attributes have to be set. For a detailed description of PI SMT, PI Point Database and their attributes please refer to the system documentation of the PI System.

6.3 PointSource Attribute

The PointSource Attribute for OPC+ IF points may be **P** for PCS7, “O” for OPC etc. This parameter assigns PI Points to the OPC+ IF. It must correspond with the parameter “PointSource” in the **SiOPC_IF.ini** file. The assignment to the particular instance of the PCS7 interface is produced via the attribute Location1.

Only values with a maximum of eight characters are supported for the PointSource. Upper /Lower case are not differentiated. Please keep in mind that some values may be occupied by the PI Server itself (e.g. “C” for Performance equations) and others may be occupied by further already existing interfaces. Please consult your PI Administrator to appoint that value.

Caution is advised if an interface used on the PI Serve supports only single character point sources. If such an interface is set to PointSource ‚A‘ for excample it will try to operate **ALL** tags wwhich have the PointSource starting with ‚A‘. This attribute is mandatory for OPC+ tags.

6.4 ExDesc attribute

6.4.1 Structure

The ExDesc attribute of a PI Point may contain several parameters. The parameters are separated by a semicolon.

<*IDENTIFIER1*=Parameter1>;<IDENTIFIER2=Parameter2>;<IDENTIFIER3=Parameter3>

For these settings upper and lower case has to be taken in account!

6.4.2 OPC ItemID (PCS7 variable name)

The first parameter always has to be the OPC ItemID. The OPC ItemID is the PCS7 respective WinCC variable name. The identifier for this parameter is “Item” (alternative “ItemID” or “Tag” may be used because of compatibility). Variable name has to be entered completely as shown in the list of variables of the data manager. This setting defines the connection of the PCS7 variable to the PI tag.

Examples:

ExDesc Entries for a PI Tag
ItemID=F41000**/**Regler.PV_IN
ItemID=OAA**/**F65101**/**VEN.FB_OPEN
ItemID=OAA**/**F65101**/**VEN.STATE;BITMASK=0x40000;SHIFT=1

This parameter is mandatory for each Point of the OPC+ IF.

6.4.3 OPC Access Path

This parameter should **not** be used in combination with a PCS7 / WinCC OPC Server.

This parameter may be added with the identifier “Path” respectively “AccessPath” and is transmitted to the OPC server. However, variables of the WinCC OPC server are fully addressed by the ItemID.

6.4.4 Bit mask and Shift parameter

With these parameters single Bits and bit groups may be selected and shifted to the right for better handling. For the functionality please refer to *chapter 5.6 Bit masking.*

The bit mask is defined with the identifier “BITMASK”. The SHIFT Function is activated with the identifier “SHIFT”.

Both parameters are optional. The identifier “SHIFT” remains ineffective without “BITMASK”.

6.5 Location1 Attribute (Instance)

This attribute defines to which instance of the OPC+ IF the point shall be assigned. Valid values are all numbers from 0 to 9.

This attribute is mandatory for all OPC+ IF points.

6.6 Location2 Attribute (Scaling on/off)

Setting this attribute “1” the gathered values are scaled. Switching scaling to on requires the attributes “UserReal1” and “UserReal2” to be set. For the functionality please refer to *chapter 5.7 Scaling.*

6.7 Location3 Attribute (IO-Type)

By this attribute the IO-Type of the point is defined. The following values are allowed:

- 1: read, time stamp provided by the OPC server with check and eventually correction (recommended!)
- 2: read, time stamp provided by the OPC server without check and correction
- 3: read, time stamp of interface PC
- 4: write, event driven „OnDataChange“.

This attribute is mandatory for all PI-CONNECT OPC+ Interface points.

Hint: In the modes 1 to 3 (read) data transmission between the OPC Server and the interface is done always only on data change. This assures a minimized load for the OPC Server and the underlying (DC-) System.

It depends on the configuration of the group cycles whether the PI API is served cyclically (recommended) or On Data Change. Please refer to chapter 5.3 Configuration of acquisition cycles. We strongly recommend to use cyclic supply of the PI API!

6.8 Location4 Attribute (Scan Class)

The number of a scan cycle defined in the file **SiOPC_IF.ini** is stated here. If the configured number is not contained in SiOPC_IF.ini the point is rejected by the interface.

This attribute is mandatory for all OPC+ IF points.

6.9 Location5 Attribute (PCS7 / WinCC Archive)

This attribute should only be set for points which will use the “archive recovery” option. Please refer to *chapter 5.9* Archive Recovery OPC HDA (optional) before setting this attribute.

In the file SiOPC_IF.ini the process value archive used by WinCC are listed und numbered serially (refer to chapter *3.5 OPC DA / OPC HDA Parameter)*. If archive recovery shall be used for a PI point set location5 to the number of the process value archive containing the archive value corresponding with the process variable. The name of the archive variable is set in the attribute “InstrumentTag”.

For systems which do not needs a process value archive name prefix in front of the archive variable name (e.g. Siemens T3000) the value -1 has to be set in lacation5 to activate the HDA function.

This attribute is mandatory for all OPC+ IF points using archive recovery.

6.10 InstrumentTag Attribute (name of the archive variable)

This attribute should only be set points intended to use “Archive Recovery”. If the attribute contains an empty string or no licence for Archive Recovery is found the function is disabled fort his point. For the functionality please refer to *chapter 5.9* Archive Recovery OPC HDA (optional).

This attribute is mandatory for all OPC+ IF points using archive recovery.

6.11 UserInt1 (Number of replacement table)

The function “replacement table” is only disposable for digital PI Points.

0 No replacement table. OPC process values without transformation into PI system.

>0 Number of replacement table.

If this function is used, it has to be proofed, that the replacement table is defined in the configuration file, before start up interface or definition of PI Points.

6.12 Userreal1 (scaling factor)

Factor of the scaling function. For the functionality please refer to *chapter 5.7 Scaling.*

The attribute is mandatory for all PI point with the attribute location2 (scaling on/off) set to “1”.

6.13 Userreal2 (scaling offset)

Offset of the scaling function. For the functionality please refer to *chapter 5.7 Scaling.*

The attribute is mandatory for all PI point with the attribute location2 (scaling on/off) set to “1”.

6.14 SourceTag Attribute

Tag name of the source point whose data should be written into the PCS7 variable. If the own value shall apply, the own tag name must be inserted. (*Refer to chapter 5.5 Writing PI data “OnDataChange”).*

This attribute is required for all points with the IO type “write on OnDataChange” (Location3 = 3). This attribute is not used for all other Points of the OPC+ IF. Please take into consideration that the indicated tag must exist in the Point Database of the PI Server.

7. Hints on PCS7, WinCC and the WinCC OPC Servers (DA and HDA)

7.1 General

The OPC DA and OPC HDA interfaces are open interfaces defined by the OPC Foundation (www.opcfoundation.org).

OPC servers provide access to process data for OPC clients. Actual values can be read or written via the OPC DA (Data Access) server. OPC HDA (Historical Data Access) servers enable access to historical process values as to process archive values.

OPC Clients make use of an OPC DA/HDA OPC Server services to read and write process data.

The Siemens OPC+ Interface is an OPC client, optimised to access the WinCC DA und HDA OPC Server. WinCC itself is a core module of a SIMATIC PCS7 Server. The SIMATIC PCS7 server may be configured redundant. This means the server software is installed on two PCs and is active on both PCs at the same time. SIMATIC PCS7 clients interact to such a redundant pair using PCS7 internal mechanisms. The WinCC OPC servers on the other hand act as two “stand alone” OPC servers providing identical data content. The Siemens OPC+ Interface is designed to access such a redundant server pair. For further information on the functionality of the redundant access refer to *chapter 5.2 Connecting to redundant PCS7 OS (redundant OPC server).*

7.2 Access to the OPC Server, Remote access via DCOM

An OPC Server exists as a COM Server. COM is an object model from Microsoft to develop distributed applications. Access takes place by operating system libraries. With the extension of COM, DCOM (distributed COM) access is possible beyond the computers. To do so it is necessary for the users logged in on the computers to grant rights to one another. Furthermore client and server PC have to be in the same domain or in the same workgroup. Please refer to the demands of the respective PCS7 version. We suggest configuring the client PC as far as possible like a PCS7 client. That is:

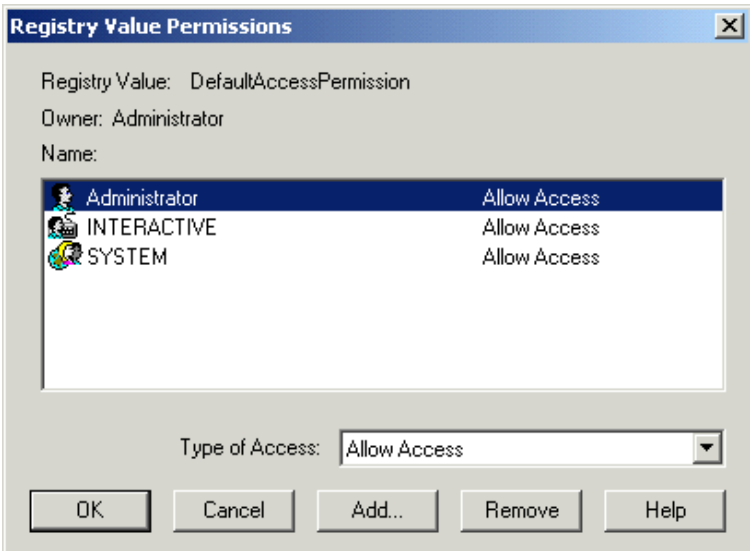
- Same operating system (including Service pack and Hot fix)
- Same operating system login and password, member of the group administrators
- Same Workgroup

Furthermore we suggest expanding the DCOM access rights on the interface PC and the PCS7 Server PCs as follows:

Start with

Start→Run→dcomcnfg

the DCOM configuration. Quit eventually occurring messages. Select the folder “Default Security” of the DCOM configuration. You will find three buttons to adjust the “Default Access Permissions”, “Default Launch Permissions” and “Default configuration Permissions”. Make sure for all 3 points the users “System”, “Interactive” and the group “Administrators” to have the maximum rights. For example here the dialog for the “Default Access Permissions”:



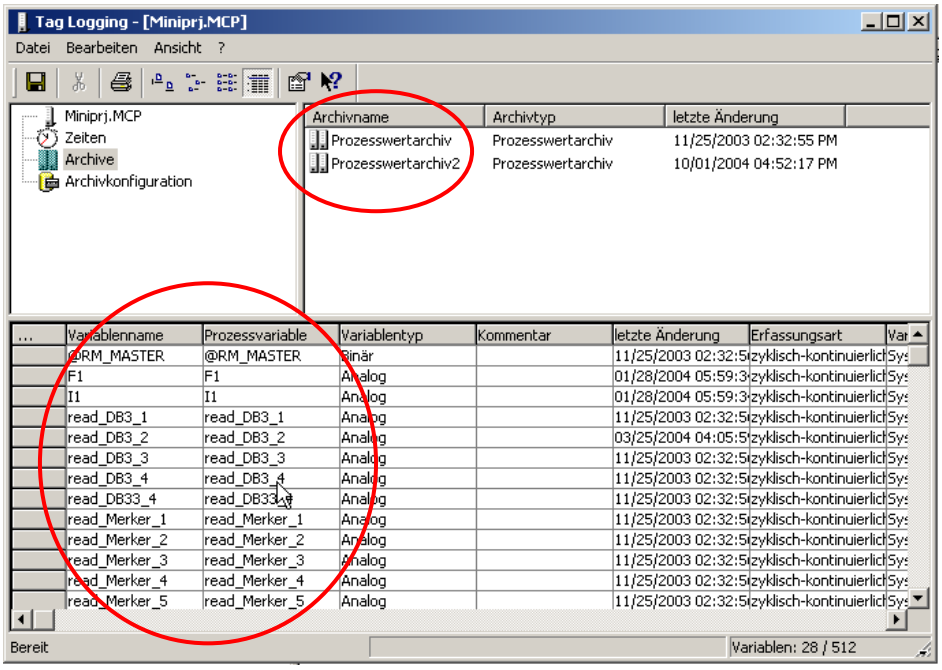
For further possibilities in configuration please refer to the OPC Foundation, the PCS7 documentation und die Help function of the Operating System.

7.3 Identification of the WinCC Process value archives

The entries for the process value archives in the file **SiOPC_IF.ini** must look as follows:

<prefix>**:**<archive name>****

The archive names may be seen directly in the WinCC Explorer's Tag Logging. The process variable name corresponding to the archive variable may also be found here.


Tag Logging - [Miniprj.MCP]
Datei Bearbeiten Ansicht ?
Miniprj.MCP
Zeiten
Archive
Archivkonfiguration
Archivname Archivtyp letzte Änderung
Prozesswertarchiv Prozesswertarchiv 11/25/2003 02:32:55 PM
Prozesswertarchiv2 Prozesswertarchiv 10/01/2004 04:52:17 PM
Variablenname Prozessvariable Variablentyp Kommentar letzte Änderung Erfassungsart
@RM_MASTER @RM_MASTER Binär 11/25/2003 02:32:5 zyklisch-kontinuierlich
F1 F1 Analog 01/28/2004 05:59:3 zyklisch-kontinuierlich
I1 I1 Analog 01/28/2004 05:59:3 zyklisch-kontinuierlich
read_DB3_1 read_DB3_1 Analog 11/25/2003 02:32:5 zyklisch-kontinuierlich
read_DB3_2 read_DB3_2 Analog 03/25/2004 04:05:5 zyklisch-kontinuierlich
read_DB3_3 read_DB3_3 Analog 11/25/2003 02:32:5 zyklisch-kontinuierlich
read_DB3_4 read_DB3_4 Analog 11/25/2003 02:32:5 zyklisch-kontinuierlich
read_DB33_4 read_DB33_4 Analog 11/25/2003 02:32:5 zyklisch-kontinuierlich
read_Merker_1 read_Merker_1 Analog 11/25/2003 02:32:5 zyklisch-kontinuierlich
read_Merker_2 read_Merker_2 Analog 11/25/2003 02:32:5 zyklisch-kontinuierlich
read_Merker_3 read_Merker_3 Analog 11/25/2003 02:32:5 zyklisch-kontinuierlich
read_Merker_4 read_Merker_4 Analog 11/25/2003 02:32:5 zyklisch-kontinuierlich
read_Merker_5 read_Merker_5 Analog 11/25/2003 02:32:5 zyklisch-kontinuierlich
Bereit
Variablen: 28 / 512

For this example the entries in the file SiOPC_IF.ini would be:

Archive_1 = local::Prozesswertarchiv
Archive_2 = local::Prozesswertarchiv2

To determine the prefix and the check the archive access we recommend to use the tool "opc_archive.exe" delivered with the interface. It is recommend to operate the tool directly on the OS Server. The EXE file may simply be copied and started fort his purpose.

Launch this program and follow the instructions on the display.

```
Programm zur Bestimmung der vorhandenen Archive bzw. Tags!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!                                                           !!!!!
!!!!!  Application require PCS7 Server performance  !!!!!
!!!!!     Do not use in critical situations.        !!!!!
!!!!!                                                           !!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Host des OPC-Servers eingeben!
Bitte '1' eingeben fuer "localhost"
>>1

ProgID des OPC-Servers eingeben!
Bitte '1' eingeben fuer "OPCServerHDA.WinCC"
>>1

Waehle maximale Anzahl die ausgelesen werden soll (empfohlen <= 100)!
Zahl eingeben (2 .. 10000)

>>20
```

Running the program the file "WinCCArchive.log" is created containing the requested archive variables including the complete path. Output should look like follows:

Info[4] <2004.10.1 17:9:2> {A} 0 : local::Prozesswertarchiv\@RM_MASTER
Info[4] <2004.10.1 17:9:2> {A} 1 : local::Prozesswertarchiv\F1
Info[4] <2004.10.1 17:9:2> {A} 2 : local::Prozesswertarchiv\I1
Info[4] <2004.10.1 17:9:2> {A} 3 : local::Prozesswertarchiv\read_DB3_1
Info[4] <2004.10.1 17:9:2> {A} 4 : local::Prozesswertarchiv\read_DB3_2
*
*

The prefix of the archive name is here "local".

7.4 PCS 7 Delta Loading

If a PCS 7 system is connected to a PI System via OPC and variables to be transmitted to the PI system were deleted in the PCS 7 data we strongly recommend not to use Delta Loading of the PCS 7 OS. This is valid as well if the variables are created again directly after deletion.

Using Archive Recovery in combination with running an OPC HDA Servers directly on the PCS 7 OS (via the WinCC Connectivity Pack or the OpenPCS 7 Client) may cause the OPC HDA Server not the recognize the newly defined archive variables in the process value archive.

Because the interface tries to do Archive Recovery starting with the latest PI archive value for new PI tags having archive recovery assigned correct writing of data to the PI archive should be checked for those tags after creation. Otherwise it may be that the OPC HDA Server does not provide data fort he new PI tags. To avoid failures the interface does not switch to OPC DA mode and removes this Variable from cyclic scanning. In this case a restart of the PCS7 OS Servers and reassigning of the tags in the interface may be necessary.

8. Hints for API Installation and PI Server Configuration

8.1 PI-interface SDK API Software installation

The interface should be used with PI SDK version tested and released by us. Please refer to chapter 1.3 Performance.

The PI-SDK Setup includes the installation of the PI-API. A detailled installion descripütion may be found in the related documentation of the PI-SDK / API of OSIsoft. The following points should be followed:

PI API buffering should be activated. Maxsize of the buffer should be set to a sufficient number.

Possibility for manual configuration:

In the file “\pipc\dat\piclient.ini” the following entries should exist:

```
[APIBUFFER]
BUFFERING=1
```

This activates PI API Node Buffering.

In older versions the additional input of **„MAXFILESIZE=2000000“** is needed in order to set the size of the data file to maximal 2GB.

The account of the PI API Node Buffering Service should be checked in the Windows service manager and adjusted. Please take care, that the service is set to “automatic”. Current versions request a user account, which is the same as the Windows account the user is logged in. Please refer to the OSISoft documentation for details of your version.

We recommend addressing the PI Server via name resolution (hosts table or DNS).

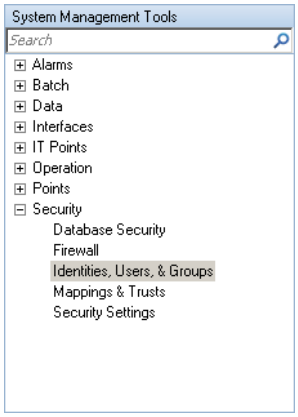
For correct function of the API buffer server it is necessary that both interface and buffer server use the same name for the PI Server und no mixing of name resolution and direct addressing (TCP-IP address) is used.

By using an high available PI Server (HA-PI) please refer to the hints in chapter 8.4 Running the Interface against a high available PI Server (HA-PI).

8.2 PI User for the PI-CONNECT OPC+ interface

To create a PI Trust account (in former times Proxy Account) a PI user and with it also the access rights are assigned to an IP address of an interface. If the user shall not be an existing user, please create a new one as follows:

Launch the System Management Tools (PI SMT) in Start->Program Files->PI System and create a new user in Security->Identities, Users & Groups with a right mouse click.



For PI server V3.3 versions:

For this start piconfig (\pi\adm\piconfig.exe) on the PI Server and execute the following:

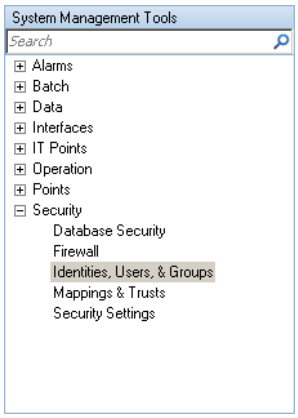
```
@table piuser
@mode create
@istr user,groups
siemens,piadmin
@ends
@exit
```

REMARK!!!: If the Interface is desired to run as PI User “piadmin”, the piconfig commands listed above are not necessary. For the Proxy Account / PI Trust Entry “piadmin” **has to be** entered instead of “siemens“.

For additional information to the commands and functioning of PIConfig refer to the manual “PI Data Archive” of OSI Software. You may use PI SMT3 to create groups and users as well.

8.3 PI-Trust entry on the PI server

To run the interface software without problems a PI trust entry is needed (in former times Proxy Account). Launch the System Management Tools (PI SMT) in Start->Program Files->PI System and create a new trust in Security->Mappings & Trusts with a right mouse click.



PITrust for PI Server V3.3:

Please configure the settings with the piconfig tool (\pi\adm\piconfig.exe) on the PI server as follows:

```
@table pitrust
@mode create
@istr trust,ipaddr,ipmask,piuser
pcs7trust, 172.19.17.12,255.255.255.255,siemens
@ends
@exit
```

The access rights of all PI Points belonging to the interface should be configured as follows (these are minimal access rights):

Interface uses the PI User “piadmin”:

Security settings since PI Server version 3.4.380: „piadmin: A(r,w) | piadmins: A(r)“

Example for older PI versions:

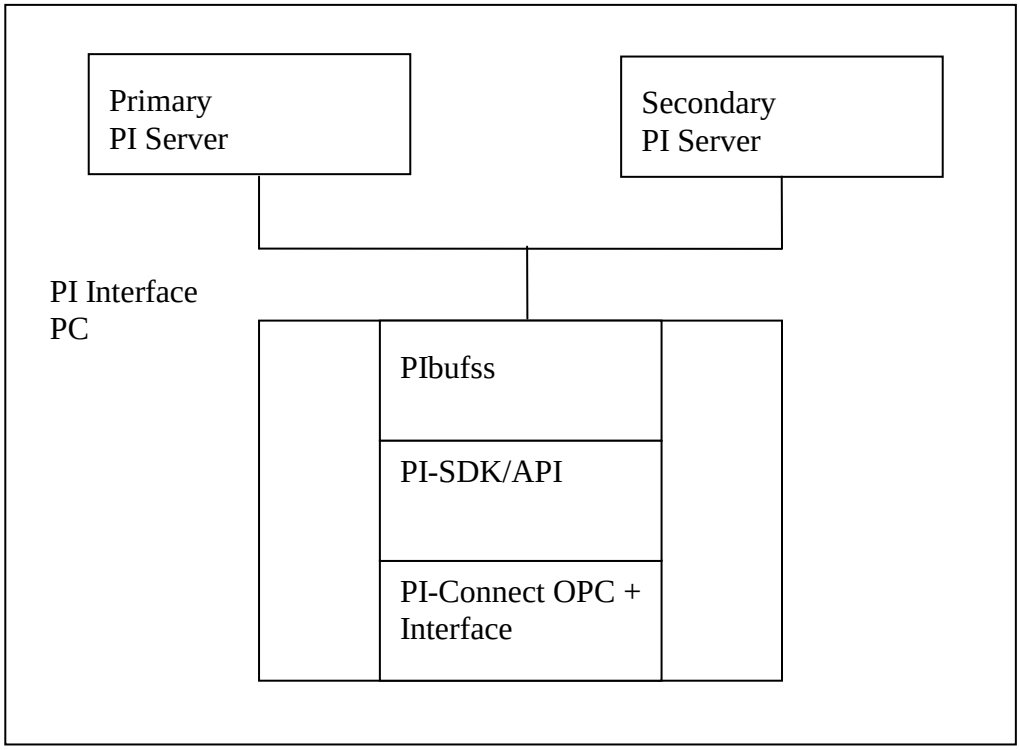
PtAccess	PtGroup	PtOwner	DataAccess	DataGroup	DataOwner
o:rw g: w:	Piadmins	Piadmin	o:rw g:r w:	piadmins	piadmin

8.4 Running the Interface against a high available PI Server (HA-PI)

The interface is tested and released with a PI Collective consisting of a Primary- and a Secondary- Server. On the interface PC the PI Buffer Subsystem has be installed and used instead of the PI Buffer Server. This configuration is as recommended by OSIsoft.

The interface software is tested successfully in this constellation with the following software releases:

- PI-Server: V3.4.380.36
- PI-Buffer Subsystem: V3.4.375.84a
- PI-SDK: V 1.3.8.387
- PI API: V 1.6.1.15



The settings in the ini files of the PI-API (Pilogin.ini, Piclient.ini) and of the SDK must be in accordance to the OSI manuals of the PI-HA Server, the PI Buffer Subsystem and the interface installation manual. It is recommended to use PI ICU to create the settings. Differing settings may cause malfunction in data transmission.

Pre-processing of the data in the PI Buffer Subsystem (Exception and Compression are done on the interface PC) and replication of the data to multiple servers may lead to higher load on the interface PC. This should be taken in account by choosing the interface PC.

Also take care of the fact that in case of connection loss to the Primary Server the interface (comparable to all OSIsoft Interfaces) does not recognize point updates and write tags (Data transfer from PI to the OPC Server) do not receive any values. After connection return the tag configuration is automatically checked and updated, the write connection returns to normal operation and point updates are recognized again.

Not supported configurations:

Replication of data using the elder PI Buffer Server service (PI-Bufserv) of the PI API to multiple autonomous PI Servers or a Collective is not tested and not supported.

The redundant version of the Interfaces is not released for operation with a PI Collective or to multiple autonomous PI Servers.

9. Revision History

Date	Author	IF-Docu	OPC+ IF	Remarks
April, 7th 2010	Hetterich	V1.3.1	V 3.6.7	Synchronizing with German Version
Aug,10th 2010	Hetterich	V1.3.2	V3.7.1	Chapter OPC Test Item enhanced Chapter Performance Tags added
Dec, 13th 2010	Hetterich	V1.3.3	V3.7.4	Interface Name PI-Connect OPC+; minor corrections
Jan, 14th 2011	Hetterich	V1.3.4	V3.7.4	Pointclass classic added
Jan, 31th 2011	Lutter	V1.3.5	V3.7.6	Revision scale function
Jul, 29th 2011	Hetterich	V1.3.6	V3.7.8	Bad State timeout
Aug, 17th 2011	Hetterich	V1.3.7	V3.7.8	Location5 value -1
Sep.2nd 2011	Hetterich	V1.3.8	V3.7.8	New License mode
Nov, 06th 2012	Bäuerle	V3.7	V3.7.10	General revision
Nov, 12th 2012	Bäuerle	V3.7a	V3.7.10	General revision
Dec, 10th 2012	Bäuerle	V3.7b	V3.7.10	paging
Dec, 21th 2012	Bäuerle	V3.7c	V3.7.10	General revision
Apr, 30th 2013	Lutter	V3.8	V3.8.0	Functional extension „HDA Window-Recovery Mode“, concretion configuration acquisition cycles and IO types, running the interface as an service
Aug, 4th 2014	Lutter	V3.8a	V3.8.5	correction of SHIFT parameter OPCStringMode 3